

# Index



## Symbols

- negative operator 258
- subtraction operator 259
- comment delimiter 347
- != not equal operator 261
- % wildcard 171, 264
- & substitution variable 333
- (+) outer join operator 65, 269
- \* multiplication operator 258
- \* SELECT all columns 27, 113
- \*/ comment end 347
- + addition operator 259
- + positive operator 258
- / division operator 259
- / execute command 315
- / SELECT termination 38
- /\* comment start 347
- < less than operator 261
- <= less than or equal operator 262
- <> not equal operator 261
- = equal operator 261
- > greater than operator 261
- >= greater than or equal operator 262

- @ command 20, 21, 313–314
- @@ command 314–315
- ^= not equal operator 261
- \_ wildcard 171, 264
- \_EDITOR variable 337
- || concatenation operator 83, 125, 166, 259

## Numerics

- 1tab\_com.sql 20, 22, 24, 256

## A

- ABS function 273
- ACCEPT command 316–317
- ACOS function 273
- ad hoc queries. *See* queries, ad hoc
- address priority ordering 178
- ADDRESS table 228
  - primary key 50
- ADDRESS\_CODES table 229
- aliases, data column 37, 39
  - COLUMN command 325
  - COMPUTE and 328

**ORDER BY** and 114, 115  
 aliases, table 68  
**ALL** operator 170, 263  
**AND** operator 173, 263  
**ANY** operator 169, 263  
**APPEND** command 317  
 arithmetic operators. *See* operators, arithmetic  
**ASCII** function 283  
**ASIN** function 274  
**ATAN** function 274  
**ATAN2** function 274  
**ATTRIBUTE** command 318  
 averages 160–161  
**AVG** function 304  
  
**B**  
 batch execution 339  
**BETWEEN** operator 170–171, 263  
**BFILENAME** function 300  
 bind variables  
   declaring 369  
   displaying values 345  
   **REFCURSOR** 369, 370  
 blocked list report 147–148  
**BREAK** command 319–320  
   **COMPUTE** coordination 320, 328, 329  
   double-spacing 136  
   grand totals 138  
   **ORDER BY** coordination 320  
   subtotals 140  
   suppressing redundant data 136  
   *See also* **CLEAR**  
**BTITLE** command 321–322  
 buffer  
   appending text 317  
   changing text 322  
   clearing 324  
   deleting lines 334  
   editing 337  
   executing commands 315, 351  
   inserting lines of text 342

  line numbers 334  
   listing 342  
   loading a file 339  
   saving to a file 351  
**BUILDING\_CODES** table 230  
  
**C**  
 calendar systems 188  
 cartesian products 160, 380  
**CATALOG** table 231  
**CEIL** function 274  
**CHANGE** command 322–323  
 character functions. *See* functions, character  
 character operators. *See* operators, character  
 character sets 186, 188, 291, 301  
**CHARTOROWID** function 290  
**CHR** function 278  
**CLEAR** command 39, 324  
 clears.sql 134  
 col\_com.sql 20, 24, 256  
 col\_num.sql 20, 21, 23, 256  
**COLLEGE\_CODES** table 232  
**COLUMN** command 37, 39, 140, 325–327  
   blocking a report 147  
   **FORMAT** option 39  
   *See also* **CLEAR**  
 columns, data  
   aliases 37, 113  
   defined 5  
   scale 26  
   width 26  
 comments, command file 346–347  
 comparison operators. *See* operators, comparison  
 compound queries  
   described 173, 266  
   requirements 266  
   *See also* set operators  
**COMPUTE** command 327–329  
   **BREAK** coordination 320, 328, 329

grand totals 138  
 subtotals 140  
*See also* CLEAR  
**CONCAT** function 279  
 concatenation described 166, 259  
 concatenation operator 166  
**CONNECT** command 330  
*See also* DISCONNECT  
 connection string 16, 17  
**CONNECT** command 330  
 constraints, table  
   described 48, 49  
   foreign key 54  
   primary key 48–49  
   referential 54  
   status of 49  
 constrnt.sql 256  
 conventions used in book x, 36  
 conversion functions. *See* functions, conversion  
**CONVERT** function 291  
**COPY** command 331–332  
 correlated subqueries  
   correlating criteria 89, 94, 385  
   described 67, 68  
   examples of 68, 87, 383, 386  
   **EXISTS** operator and 88  
   join errors and 68  
   query populations and 87, 99  
**COS** function 274  
**COSH** function 274  
**COUNT** function 117, 304  
   null values and 117  
**COUNT (\*)**  
   examples of 83, 85, 87, 89  
   null values and 303  
   population counts and 81  
**COUNT (DISTINCT . . .)**  
   examples of 83, 85, 87, 89  
   population counts and 81  
 craft, SQL ix–x, 4, 76, 113  
 crosstabulations. *See* two-way crosstabula-

tions  
 currency symbols 188  
 cursor bind variables 369–370

## D

data dictionaries. *See* dictionaries, data  
 data tables. *See* tables, data  
 data values described 5  
 data views. *See* views, data  
 database objects. *See* objects, database  
 datatypes  
   **DATE** defined 26  
   examples of 25  
   **LONG** 332  
   **NUMBER** defined 26  
   **OBJECT TYPES** 318  
   **VARCHAR2** defined 26  
 date arithmetic 124, 126  
**DATE** datatype. *See* datatypes  
 date functions. *See* functions, date  
 date.sql 135  
 date-stamping a report 135  
**DECODE** function 299  
   described 125  
   examples of 124, 157  
**DEFINE** command 332–333  
**DEGREE\_CODES** table 233  
**DEL** command 334  
**DEMOG** table 234  
**DESCRIBE** command 27, 335  
 dictionaries, data  
   described 25  
   examples of 25, 33, 53, 54  
   joins and 50  
**DISCONNECT** command 336  
*See also* CONNECT  
**DISTINCT**, in group functions 207, 303  
**DISTINCT**, in **SELECT** clause 113  
 double-spacing a report 136, 320  
 downloads 20  
 driving table 74  
   examples of 80, 93

DUMP function 299  
duplicates, suppressing 320

## E

EDIT command 40, 336–337  
editing, command file 336  
editor, changing default 337  
EMPTY\_BLOB function 298, 300  
EMPTY\_CLOB function 300  
end-user reporting tools 14  
equijoin  
    described 51  
    examples of 51, 55, 82  
    INTERSECT equivalence 267  
errors, operating system 371  
errors, Oracle  
    ORA-00900 340  
    ORA-00911 340  
    ORA-00923 347  
    ORA-00932 167  
    ORA-00979 119  
    ORA-01722 167  
    ORA-01799 69  
    ORA-01858 167  
    See also SHOW command  
errors, SQL 372  
ESCAPE option. See LIKE operator  
EXECUTE command 337–338  
executing  
    buffer command 351  
    command file 313, 314, 363  
    operating system command 341  
    PL/SQL command 337  
    SQL command 315  
EXISTS operator 88, 90, 171, 263  
    examples of 94, 385  
    NOT variation 90, 263  
    query performance and 90  
    query populations and 88, 94  
EXIT command 17, 338–339  
exit SQL\*Plus 338  
EXP function 275

EXPLAIN PLAN 269  
expressions described 34  
extracts  
    delimited 150–151, 393  
    fixed-format 152, 394

## F

falling through a join 64–66  
    consequences 62, 64  
    described 62, 381  
    examples of 63  
    one-to-none relationship 62  
    outer joins and 64  
    population counts 75, 81  
    solutions for 64  
file names  
    SQL program 40  
    See also SET SUFFIX 337  
fill mode (FM) format modifier 292, 294  
fk.sql 54, 256  
FLOOR function 275  
FM fill mode format modifier 292, 294  
fn.sql 256  
footers, creating report 347  
foreign keys 52–56  
    joins and 46, 52, 216  
    See also joins, table  
    See also referential constraints  
format models  
    date elements 200–202, 294–296  
    date examples 198  
    fill mode (FM) modifier 204, 294  
    number elements 202–204, 296–298  
    ROUND date elements 196, 288  
    TO\_CHAR date function 125  
    TRUNC date elements 196, 288  
formfeeds 358  
frequency distributions. See one-way frequency distributions  
FROM clause 34  
    driving table 74  
    subquery 146, 158, 162, 396

## functions

- categories of 272
- character 191–194, 277–284
- conversion 197–204, 289–298
- date 194–196, 284–288
- group 207–208, 303–305
- miscellaneous 204–205, 298–303
- number 189–191, 272–277
- SELECT** clause and 111
- See also* names of functions
- functions, user created 111

**G**

- GET** command 339–340
- GRADE** table 235
- GRADE\_CODES** table 236
- grand totals, report 138, 328, 329
- GREATEST** function 300
- GROUP BY** clause 35, 116–120
  - examples of errors 120
  - SELECT** coordination 118
- group functions. *See* functions, group

**H**

- HAVING** clause 35, 75, 120–122
- headers, report 349
- headings, column
  - COLUMN** command 325
  - disabling 147
  - justification of 141
  - multiple lines 326, 360
  - TTITLE** use 150
- HELP** command 340–341
- help, SQL\*Plus 340
- HEXTORAW** function 291
- hierarchical queries 112, 176, 269
- HOST** command 341

**I**

- if logic. *See* **DECODE** function
- IN** operator 168, 262
  - examples of use 63, 80, 169

**NOT** variation 262

- indexes 49
- indexes.sql 256
- INITCAP** function 279
- INPUT** command 342
- INSTR** function 193, 283
- INSTRB** function 283
- INTERSECT** operator 175, 267
- IS NULL** operator 264

**J**

- join errors
  - described 62–63
  - falling through 64–65
  - multiplying through 66–67
  - simultaneous 69–70
  - solutions for simultaneous 70, 102
- joins, table 45–59
  - arithmetic of 56–59
  - equijoin 51
  - examples of 51, 55, 94
  - foreign keys used 46
  - multiplicative effect 58, 59
  - need for 6–7, 10
  - outer join 64
  - primary keys used 46, 49
  - stepping through 75
  - testing 75, 99–102
  - See also* cartesian products
  - See also* equijoins
  - See also* falling through a join
  - See also* foreign keys
  - See also* join errors
  - See also* multiplying through a join
  - See also* outer joins
  - See also* primary keys
  - See also* stepping through a join
- Julian dates 293

**K**

- Komenda College
  - described 18

OR3 owner 21  
 registration database 21–26  
 registration process 91  
 simulation x, 19

**L**

languages, procedural 13  
 languages. *See* National Language Support 186  
 LAST\_DAY function 285  
 LEAST function 300  
 LENGTH function 283  
 LENGTHB function 283  
 LEVEL pseudocolumn 112  
 LEVEL\_CODES table 237  
 LIKE operator 171–172, 264  
 line numbering 350, 361  
 linesize, report. *See* SET LINESIZE  
 linguistic sort sequence 280, 284  
 LIST command 342–343  
 list reports 146  
 listing a command file. *See* LIST command  
 literals described 89  
 LN function 275  
 local databases 16  
 LOG function 275  
 logical operators. *See* operators, logical  
 login 16, 17  
 LONG datatype. *See* datatypes  
 LOWER function 279  
 LPAD function 150, 279  
 LTRIM function 279

**M**

MAJOR\_CODES table 54, 238  
   primary key 54  
 master-detail list 153  
 MAX function 68, 304  
 Mills, C. Wright ix  
 MIN function 304  
 MINUS operator 176, 268  
 MOD function 275

MONTHS\_BETWEEN function 285  
 multiplying through a join 66–67  
   consequences 62  
   correlated subqueries and 67  
   debugging with HAVING 121  
   examples of 63  
   one-to-many relationships 62  
   population counts 75, 81  
   primary keys and 66  
   ROWID if no primary key 112  
   solutions for 67

**N**

NAME table 25, 239  
   primary key 49  
 NATION\_CODES table 240  
 National Language Support 186–189  
 NEW\_TIME function 286  
 NEXT\_DAY function 286  
 NLS\_CALENDAR 188  
 NLS\_CHARACTERSET 188  
 NLS\_CHARSET\_DECL\_LEN function 301  
 NLS\_CHARSET\_ID function 301  
 NLS\_CHARSET\_NAME function 301  
 NLS\_CURRENCY 188, 292  
 NLS\_DATE\_FORMAT 188  
 NLS\_DATE\_LANGUAGE 188, 292  
 NLS\_INITCAP function 280  
 NLS\_ISO\_CURRENCY 188, 292  
 NLS\_LANGUAGE 187, 294  
 NLS\_LOWER function 280  
 NLS\_NCHAR\_CHARACTERSET 188  
 NLS\_NUMERIC\_CHARACTERS 188  
 NLS\_NUMERIC\_CURRENCY 292  
 NLS\_SORT 188, 280, 284  
 NLS\_TERRITORY 187  
 NLS\_UPPER function 280  
 NLSSORT function 284  
 normalization  
   described 6  
   effects of 10, 46  
 NOT operator 173, 266



- described 48, 256
  - examples of use 49, 50, 53, 54
  - `PLAN_TABLE` table 269
  - populations, query
    - baseline counts 74, 80, 83, 85, 87, 89
    - `COUNT(DISTINCT . . .)` and 81
    - described 74, 77, 79
    - examples of 79–80, 83, 85, 87, 89
  - positional notation. *See* `ORDER BY`
  - `POWER` function 275
  - primary keys 46–52
    - candidates for 47, 50
    - constraints created 48
    - institutional policies and 51–52, 379
    - join errors and 66
    - joins and 46, 49
    - population counts and 83
    - See also* joins, table
    - See also* names of tables
    - See also* `pk.sql`
  - `PRINT` command 345
  - `printer.sql` 134
  - `PRIOR` operator 176, 269
  - procedural languages. *See* languages, procedural
  - `PROMPT` command 346
    - See also* `ACCEPT` command
  - prompting users 316, 344, 346
- Q**
- queries, ad hoc 92–95, 211
  - query construction
    - clarification process 93
    - examples of 32–34, 93–95
    - process described ix, 32, 36–37, 74
    - steps, detailed 74
    - top-down fallacy 74
    - Who, What, Where 74
    - See also* What functional area
    - See also* Where functional area
    - See also* Who functional area
  - query performance. *See* performance, query
- R**
- query
    - query populations. *See* populations, query
  - `RAWTOHEX` function 291
  - referential constraints 54
    - error trapping and 55
  - `REG` table 242
  - `REG_STATUS_CODES` table 243
  - registration, Komenda 91
  - relational databases
    - features of 46
    - history 12
  - `REMARK` command 346–347
  - remote databases 16
  - `REPFOOTER` command 135, 347–349
  - `REPHEADER` command 135, 349–351
  - `REPLACE` function 280
  - reporting solutions
    - database objects 11
    - end-user tools 14
    - procedural languages 13
    - statistical tools 13–14
  - reserved words
    - conventions x
    - described 36
  - `ROOM_CODES` table 244
  - `ROOM_TYPE_CODES` table 245
  - `ROUND` date function 195, 286
  - `ROUND` number function 190, 276
  - `ROWID` pseudocolumn 112
  - `ROWIDTOCHAR` function 291
  - `ROWNUM` pseudocolumn 112
    - examples of 389
  - rows described 5
  - `RPAD` function 153, 281
  - `RTRIM` function 281
  - `RUN` command 351
    - See also* `/` command
- S**
- `SAVE` command 351–352

- saving
  - query output 362
  - SQL program files 40
  - SQL\*Plus system variables 364
- scale, **NUMBER** columns 26
- SCHEDULE** table 246–247
  - primary key 94
- SELECT** clause 111–114
  - \* symbol 27
  - column order in reports 37, 38
  - described 34
  - DISTINCT** 113
  - GROUP BY** coordination 118
  - population counts 74
  - structuring 114
  - syntax 111
- SELECT** command
  - clauses 34–35
  - termination character 38
  - See also* names of clauses
- semicolon, **SELECT** termination 38
- SET** command 352–360
  - ECHO** 352
  - LINESIZE** 358
  - NEWPAGE** 358
  - PAGESIZE** 151, 358
  - PAUSE** 28, 105, 358
  - SUFFIX** 337
  - TERMOUT** 358
  - TIMING** 105, 359
  - TRIMSPPOOL** 151
- set operators. *See* operators, set
- shell formatting program 131–135, 218
- SHOW** command 360–362
  - See also* **SET** command
- SIGN** function 191, 276
- SIN** function 276
- SINH** function 276
- snapshot tables 10
  - See also* static tables
- SOME** operator 169, 263
- sorting
  - character sets and 115, 186, 291
  - linguistic sort sequence 280, 284
  - NLS\_SORT** 280, 284
  - See also* **ORDER BY**
- SOUNDEX** function 281
- source.sql 256
- SPPOOL** command 132, 362–363
- SQL**
  - program file names 40
  - reporting advantages 13
  - structured queries 41–42
  - syntax, mixed case 36
  - uses for 13
- SQL\*Plus**
  - command line access 16
  - command syntax 311–313
  - commands listed 309
  - conventions used in book 308
  - environment, altering 352
  - exiting 338
  - formatting reports 76
  - help system 340
  - login 16–18
  - system variables 27
  - uses for 13, 309
- SQLRT** function 276
- START** command 20, 21, 363–364
  - See also* @ command
- STATE\_CODES** table 248
- static tables 10, 12
- statistical tools 13–14
- STDDEV** function 304
- stepping through a join 75, 99–102
  - examples of 100–102, 104–108
- STORE** command 364–365
- structured SQL queries 41–42
  - benefits of 41
  - clauses, order of 36
  - examples of 41
  - suggestions for 41
  - Who, What, Where 42–43
- STUDENT** table 53, 249–250

primary key 53  
 STUDENT\_STATUS\_CODES table 251  
 SUBJECT\_CODES table 252  
 subqueries described 67  
 substitution variables  
   described 333  
   escape character and 359  
 SUBSTR function 153, 192, 281  
 SUBSTRB function 281  
 subtotals, report 139, 328, 329  
 SUM function 305  
 SYS owner 21  
 SYSDATE function 124, 286  
 SYSTEM owner 21  
 system variables 350  
   defined 27  
   showing 360  
   SQL.PNO 132  
   *See also* SHOW command

## T

tabl\_com.sql 20, 22, 23, 256  
 tabl\_num.sql 20, 21, 22, 256  
 tabldict.sql 20, 25, 256  
   output file 26  
 table joins. *See* joins, table  
 table names  
   conventions in book x  
 tables, data  
   aliases 68  
   code tables, associated 52  
   described 5  
   reporting advantages 11  
   *See also* names of tables  
 TAN function 276  
 TANH function 277  
 TERM\_CODES table 253  
 terminal output  
   pausing 27–28  
   suppressing 358  
 time component of dates 126, 195  
 time zones 287

time.sql 135  
 time-stamping a report 135  
 TIMING command 365–366  
   *See also* SET TIMING  
 timing statistics 359, 365  
 titles, report. *See* BTITLE or TTITLE  
 TO\_CHAR function, dates 124, 125, 198, 292  
 TO\_CHAR function, numbers 198, 292  
 TO\_DATE function 199, 293  
 TO\_MULTI\_BYTE function 293  
 TO\_NUMBER function 199, 293  
 TO\_SINGLE\_BYTE function 294  
 TRANSLATE function 282, 294  
 TRUNC date function 195, 287  
 TRUNC number function 124, 126, 190, 277  
 truncating data values in reports 141  
 TTITLE command 132, 366, 366–368  
   report headings 150  
 two-way crosstabulations 157, 395  
 two-way reports 156

## U

UID function 302  
 UNDEFINE command 368  
   *See also* DEFINE command  
 UNION ALL operator 175, 267  
 UNION operator 175, 267  
   examples of use 177–181  
 UPPER function 282  
 user defined variables 332, 368  
 USER function 302  
 USERENV function 302  
 username, Oracle  
   described 16  
   object owner 21  
   USER function 302  
 utility programs  
   1tab\_com.sql 20, 256  
   col\_com.sql 20, 256  
   col\_num.sql 20, 256

constrnt.sql 256  
 download site 20  
 fk.sql 54, 256  
 fn.sql 256  
 indexes.sql 256  
 listed 20  
 pk.sql 48, 256  
 source.sql 256  
 tabl\_com.sql 20, 256  
 tabl\_num.sql 20, 256  
 tabldict.sql 20, 256  
*See also* names of programs

described 32, 79  
 examples of 37  
*See also* populations, query  
*See also* query construction  
 width, data column 26  
 wildcards  
     *See* LIKE operator  
 wrapping data values 140, 217, 326  
 www.komenda.com 20

**V**

V1\_NAME\_ADDRESS 33  
 VARCHAR2. *See* datatypes  
 VARIABLE command 369–370  
     *See also* PRINT command  
 VARIANCE function 305  
 views, data 10  
     reporting advantages 10, 11  
     V1\_NAME\_ADDRESS 33  
 VSIZE function 302

**W**

warehouse tables 10  
     *See also* static tables  
 What functional area  
     clauses used 42  
     described 32  
     examples of 38  
     *See also* query construction  
 WHENEVER OSERROR command 371–372  
 WHENEVER SQLERROR command 372–373  
 WHERE clause 35, 37  
 Where functional area  
     clauses used 42  
     described 32  
     examples of 82  
     *See also* query construction 74  
 Who functional area  
     clauses used 37, 42













